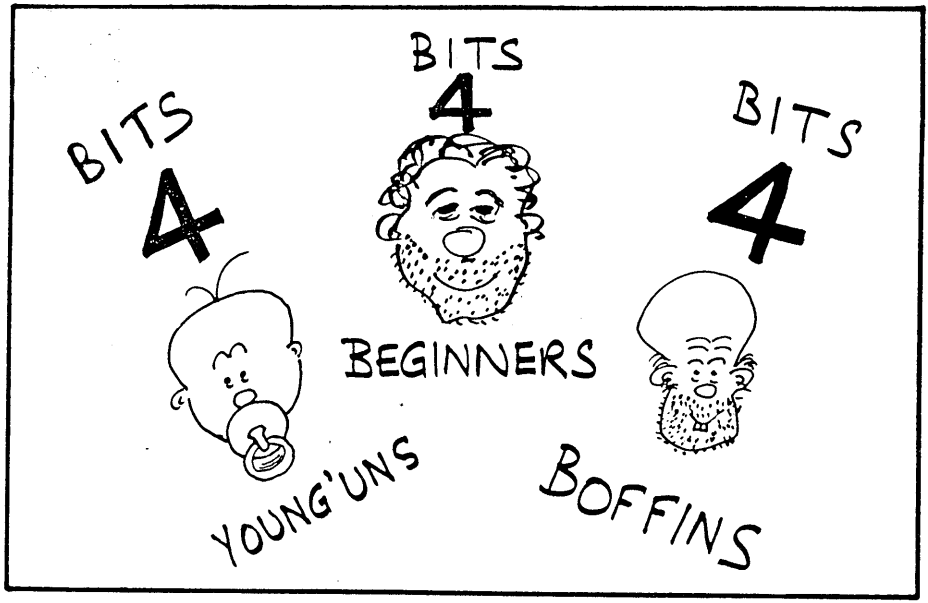
INTERNATIONAL TI-LINES

Volume 3, Issue 1 June 1st., 1986

A PUBLICATION OF THE INTERNATIONAL TI USER GROUP



Formerly OXON TI USERS

A PUBLICATION OF THE INTERNATIONAL TI USER GROUP

Tel. (02357) 2360
WESSEX PRESS
9-11 Mill Street, Wantage, Oxon OX12 9AB.

PETER G. Q. BROOKS
96 BANBURY ROAD
OXFORD OX2 6JT

Index to articles Page

BALDIE'S BURBLE....3
The replacement for the late Editorial
BITS 4 YOUNG 'UNS....6
This month it's GA0002, otherwise known as AWARI, by GENE HITZ
CONTACTS....9
Including a special Contact for subscribers in Ireland
BITS 4 BEGINNERS....10
Manipulating graphics definition strings
PROGRAMMER'S NOTES....12
Based on observations by JOHN ROE
BITS 4 BOFFINS....14
The TI BASIC Symbol Table, by BOB DILWORTH from NEW HORIZONS
LETTERS....18
JOHN RICE and JOHN ROE with confirmed faults in FUNNELWRITER, an offer from JIM BOUDIER who has a TEX-COMP catalogue, and details from DAVE HEWITT with regard to the 32K RAM Expansion
COMMENTS....20
COLIN HINSON with comments on a variety of matters which saw the light of day in previous issues of TI-LINES
A LETTER TO THE TI-99/4A USER GROUPS....24
From JIM PETERSON of TIGERCUB SOFTWARE
PROJECT NOTES....26
4 your in-4-mation, follow-up to the New Horizons RAMDISK project notified last month
ADVERTISEMENT....27
4FRONT from NEW DAY COMPUTING, the magazine on disk and tape
MODEM REVIEW....28
The GEC Datachat from GORDON PITT, West Midlands TI Users
STILL AVAILABLE FROM TI....31
Details from COLIN HINSON of the current availability of some documentation from TI
BULLETIN BOARD....32
More sales and wants

INTERNATIONAL TI-LINES: A PUBLICATION OF THE INTERNATIONAL TI USER GROUP

Magazine scanned 2022 by Stephen Shaw

B A L D I E ' S B U R B L E

To all those who have resubscribed to this third volume of monthly, well almost monthly, scrambled brain dump, welcome and have your eyes managed to recover yet ?

Only the most unobservant could fail to notice the slight changes made to TI-LINES, and they are all in response to the filled questionnaires which almost all of you have returned (all of you who resubscribed, that is).

For the first time in over two years I have been able to examine a quite detailed response from members, showing some surprising views, and providing me with the invaluable feedback which should help me to alter TI-LINES to your design.

I have to be honest and admit that one or two things did knock me back on my heels - for example, the fact that many "boffins" indicated that there was a tendency for too much emphasis on boffin-type data and not enough for the beginner.

Another was the strong feeling that the Editorial was too long, and in fact these two particular findings have nicely balanced each other: I can increase the amount for beginners, keep the "Boffinarium" (which I suspect is still wanted by the boffins despite the questionnaires!), and curb the Editorial.

In fact, there are other, perhaps more important, changes which I am in the process of making.

The names of the group and of the newsletter are changing, both to indicate the strong international flavour and to try and escape the rather parochial tag which had (unfairly) become attached to OTIU. In many instances, OTIU was referred to either as the Oxford TI User group or even as TI-Lines, and although OTIU was both national and international in structure, another group (who shall remain nameless to protect the slightly naughty) kept advertising itself as the "sole national UK group", making OTIU seem restrictive and restricted - unfair and untrue.

However, Trade Descriptions Act aside, THE INTERNATIONAL TI USER GROUP is now OTIU's banner, more accurately reflecting its make-up, and the newsletter is now INTERNATIONAL TI-LINES, also reflecting its make-up.

If, however, YOU feel that the change has been for the worse and not the better, make your voice heard (but not by dropping your subscription, pretty please!) and offer constructive criticism so that I can be helped to get it right.

INTERNATIONAL TI-LINES (IT - pronounced Eye Tea, or just It) will settle into a permanent format once I have sorted out my new printer (which I could not afford but I need to improve some of the graphic imaging).

)))))))

There is a new structure to the newsletter, dividing up into specialist sections. The beginner is catered for by BITS 4 BEGINNERS, while the boffin/intrepid explorer is provided with continued brain-boggling in the Boffinarium, or BITS 4 BOFFINS.

Where a boffinistic article is likely to be too big to fit nicely into IT, I will publish it as a separate IT SUPPLEMENT at a reasonable price, so that eager beavers can gorge themselves on wood pulp covered in melted plastic and carbon particles.

Another novelty will be the possibility of getting THE BETTER OF IT - or selections from past issues of TI-LINES and from IT in the future. This will merge with the QUINSOFT BOOKLETS SERIES which is still under way.

Younger members of ITUG (Eye Tug - well, it's less of a mouthful than OTIU) who might like to practise their keyboard bashing (and also those somewhat older who have a similar inclination) are offered a listing a month from the former TIHOME SOFTWARE COLLECTION in BITS 4 YOUNG 'UNS.

Don't knock it - I cut my teeth on looking at listings and spotting other programmers' good and bad bits. The best way to learn BASIC is to practise by imitation. Once you have some of the language learned by repetition, you can go on to develop your own (inimical ?) style.

You might also consider going on to learn C99, a version of Small C, which is very BASIC-like in its language and structure, but which can give you power only Forth and pure Assembly Language could otherwise provide.

I'm hoping for some Public Domain Small C, and for a series for BASIC buffs wanting to convert.

The Editorial is not really dead - it lives on in this, Baldie's Burble, which will still let me try to extend your horizon beyond the 4A alone, but within a few, rather than several, pages. It gives me the chance to offer you a 4 TASTE (yes, I know, it's getting tacky, but can you blame me ?) of what is to come.

I will be prepared to offer a FREE SUBSCRIPTION to ITUG and IT to anyone who is prepared to undertake a REGULAR, good quality, feature of their own choosing. First in this category is SCOTT COPELAND and his alter ego, JO ANN, who will hopefully be covering an ADVENTURE HINTS feature and possibly also a BEGINNERS' ASSEMBLY LANGUAGE TUTORIAL.

Few respondents indicated that they would be likely to write even a small article for IT, so as usual they will have to rely on the willingness of the more courageous to put pen to paper (or similar). I can only do so much, and unless I get on with other literature I am going to lose the audience and the market, which will please nobody.

IT will run to about 32 sides maximum, and I have ordered another printer (which I cannot afford!) in order to try and tart up IT's appearance. This will inevitably mean changes in typeface, but as some of you have complained about the legibility of the small OKI MICROLINE characters this can only be a step in the right direction.

If anyone fancies their hand at cartooning, feel free - the title page will contain anything which either relates to the content of an issue, or which has a 4A flavour.

))))))

I have recently received a copy of 4FRONT, the tape/disk magazine from HARRY PRIDMORE of NEW DAY COMPUTING. Great minds think alike, etc., but I have just spent a week working out all sorts of 4 type puns and along comes 4FRONT with a similar tone to its presentation! I just hope that I haven't copied any of Harry's titles (apologies if I have and I will do my best to avoid NAME CONFLICTS in future). You will find elsewhere an advert for New Day, and I can recommend 4FRONT to you. I have only had chance to run the disk-based version (slightly different from the tape-based magazine, so why not be a greedy pig and get both ?) but I found it very interesting. Old diehards like IAN MARTIN and NEIL LAWSON appear as contributors, and you will also find SCOTT COPELAND (see earlier) as well. Our own DAVE HEWITT has his computer-aided circuit design program in this first issue, and it looks as though 4FRONT will cater for a wide range of tastes.

A tremendous amount of work must have gone into the disk magazine (or, as I believe is fashionable these days, "diskazine"); either way, I know how much effort is required for IT's production and 4FRONT must need ten times that amount!

To quote a lady from a past TV age (and thus show my own age):

"Oi'll give it foive!"

ITUG also has a new aspect to its membership, started during the second volume of TI-LINES by GORDON PITT and his WEST MIDLANDS TI USERS, and that is block subscription by members of independent, democratically-run User groups. (Remember that ITUG is a professional service for TI owners world-wide, and is thus not a club.)

This opening issue of IT greets the members of MALTBY TI USERS, headed by MARK LEE, whose efforts to publicise his group locally are an object lesson in what can be done and how to go about it. I hope shortly to publish cuttings from the press which Mark has received, and I have taken a few pointers myself for use in spreading the word about IT.

(As in: Are You Desperate For IT ? Or, Get IT While It's Hot!)

In addition to the resubscriptions, and along with members of the Maltby group, we say Welcome To IT to quite a small army of TI buffs:

Mr HAYWARD, Mr HINSON, Mr BELL, Mr COPELAND, Mr BROWN, Mr LARKIN, Mr O'CONNOR, Mr MacMAHON, Mr CLOUD, Mr GREENWOOD, Mr SCOTT, Mr FREW, Mr GILHOOLY, Mr HAIGH, Mr PARKES, Mr BUNKER, Mr DOUGLASS, Mr PRIDMORE, to name but a few.

We hope that you manage to get what you want out of IT, if you'll pardon these endless puns...

I'm nearing the end of the section which I have allocated to myself for this issue, and also nearing the end of my resources, as it is 5.25 am on Tuesday June 3rd., and while I am trying to get IT out (all right, all right, calm down) by this Saturday, I still find an odd tendency to need to enter an altered state of consciousness - i.e., sleep!

I hope that you will approve of the alterations that I have made - bear in mind that some of your preferences are responsible - and I hope that you will give IT a chance.

))))))

```
100 REM TSC GA0002
110 REM GENE HITZ WAUWATOSA,
WI. 53213
120 CALL CLEAR
130 CALL COLOR(16,11,11)
140 CALL CHAR(152,"")
150 CALL CHAR(113,"6D6FFF7FF
FFF793F")
160 CALL CHAR(114,"000101030
71E3C7C")
170 CALL CHAR(115,"7FC773618
11F3F7F")
180 CALL CHAR(116,"FF7F7C3E3
F3F3F3F")
190 CALL CHAR(117,"EFC707070
783C3E3")
200 CALL CHAR(118,"1F0F0F070
3030101")
210 CALL CHAR(119,"E3F37B7F3
F9F9FCF")
220 CALL CHAR(120,"010000010
1037F7E")
230 CALL CHAR(121,"CFE7E773F
0F8F838")
240 CALL CHAR(122,"B6F6FFFEF
FFF9EFC")
250 CALL CHAR(123,"008080C0E
0783C3E")
260 CALL CHAR(124,"FEE3CE868
1F8FCFE")
270 CALL CHAR(125,"FFFE3E7CF
CFCFCFC")
280 CALL CHAR(126,"F7E3E0E0E
0C1C3C7")
290 CALL CHAR(127,"F8F0F0E0C
0C08080")
300 CALL CHAR(128,"C7CFDEFEF
CF9F9F3")
310 CALL CHAR(129,"800000808
0C0FE7E")
320 CALL CHAR(131,"")
330 CALL CHAR(130,"F3E7E7CE0
F1F1F1C")
340 GOSUB 370
350 GOSUB 590
360 GOTO 680
370 CALL CLEAR
380 CALL HCHAR(16,15,131)
390 CALL HCHAR(16,18,131)
400 CALL HCHAR(16,16,113)
410 CALL HCHAR(17,15,114)
420 CALL HCHAR(17,16,115)
430 CALL HCHAR(18,15,116)
440 CALL HCHAR(18,16,117)
450 CALL HCHAR(19,15,118)
460 CALL HCHAR(19,16,119)
470 CALL HCHAR(20,15,120)
480 CALL HCHAR(20,16,121)
490 CALL HCHAR(16,17,122)
500 CALL HCHAR(17,18,123)
510 CALL HCHAR(17,17,124)
520 CALL HCHAR(18,18,125)
530 CALL HCHAR(18,17,126)
540 CALL HCHAR(19,18,127)
550 CALL HCHAR(19,17,128)
560 CALL HCHAR(20,18,129)
570 CALL HCHAR(20,17,130)
580 RETURN
590 FOR Y=1 TO 3
600 CALL SOUND(100,110,0,111
,1,-5,8)
610 CALL SOUND(400,110,30)
620 FOR X=1 TO 3
630 CALL SOUND(100,110,2,-5,
15)
640 CALL SOUND(400,110,30)
650 NEXT X
660 NEXT Y
670 RETURN
680 PRINT TAB(12);"AWARI":
690 PRINT TAB(8);"'PROGRAM I
NNOVATORS'":TAB(12);"NO. 127
":
700 GOSUB 590
710 FOR X=1 TO 100
720 NEXT X
730 FOR X=9 TO 13
740 CALL COLOR(X,2,16)
750 NEXT X
760 PRINT "AWARI IS AN ANCIE
NT AFRICAN GAME PLAYED WITH
36 SEEDS,  6 PITS ON EACH SI
DE WITH AN"
770 PRINT "ADDITIONAL HOME P
IT ON EACH END.":"A MOVE IS
MADE BY TAKING ALLTHE SEEDS
FROM ANY PIT ON   YOUR SIDE.
"
780 PRINT "THESE SEEDS ARE S
OWED ONE INEACH PIT AROUND T
HE BOARD   COUNTER-CLOCKWIZE
"
790 INPUT B$
800 CALL CHAR(96,"")
810 CALL CHAR(97,"0000000000
004")
820 CALL CHAR(98,"0000000040
004")
830 CALL CHAR(99,"0000400040
004")
840 CALL CHAR(100,"400040004
0004")
850 CALL CHAR(101,"400040004
0005")
860 CALL CHAR(102,"400040005
0005")
```

TSC PROGRAM LISTING

>>>

```
870 CALL CHAR(103,"400050005
0005")
880 CALL CHAR(104,"500050005
0005")
890 CALL CHAR(105,"500050005
00054")
900 CALL CHAR(106,"500050005
40054")
910 CALL CHAR(107,"500054005
40054")
920 CALL CHAR(108,"540054005
40054")
930 CALL CHAR(109,"540054005
40055")
940 CALL CHAR(110,"540054005
50055")
950 CALL CHAR(111,"540055005
50055")
960 CALL CHAR(112,"550055005
50055")
970 CALL COLOR(14,5,16)
980 CALL COLOR(15,9,16)
990 CALL CHAR(136,"183C7EFF1
8181818")
1000 CALL CHAR(144,"18181818
FF7E3C18")
1010 PRINT "IF THE LAST SEED
 OF YOUR     MOVE IS SOWED IN
 YOUR HOME   PIT, YOU GET A S
ECOND TURN."
1020 PRINT "IF THE LAST SEED
 IS SOWED IN AN EMPTY PIT, D
IRECTLY      OPPOSITE AN OCCU
PIED PIT,"
1030 PRINT "ALL THE SEEDS IN
 THAT        OPPOSITE PIT PLU
S THE LAST   SEED, ARE MOVED
TO THAT      PLAYERS HOME PIT
."
1040 PRINT "WHEN EITHER SIDE
 IS EMPTY,   THE GAME IS FINI
SHED. THE    PLAYER WITH THE
MOST SEEDS   IS THE WINNER."
1050 PRINT "USE ARROWS TO IN
DICATE MOVE."
1060 INPUT B$
1070 PRINT "YOUR OPPONENT WI
LL BE        'UMBALA KUBUI',
LEGENDARY    CHAMPION OF CENT
RAL AFRICA."
1080 U=11
1090 V=19
1100 W=9
1110 Z=9
1120 DATA 0
1130 FOR X=1 TO 1000
1140 NEXT X
1150 DIM B(13),G(13),F(50)
1160 PRINT : :
1170 INPUT "YOUR NAME PLEASE
 ? ":A$
1180 READ N
1190 E=0
1200 CALL CLEAR
1210 FOR I=0 TO 12
1220 B(I)=3
1230 NEXT I
1240 C=0
1250 F(N)=0
1260 B(13)=0
1270 B(6)=0
1280 INPUT "DO YOU WISH TO G
O FIRST  ":B$
1290 CALL COLOR(3,2,16)
1300 CALL COLOR(4,2,16)
1310 IF SEG$(B$,1,1)="N" THE
N 1340
1320 GOSUB 2220
1330 GOTO 1370
1340 GOSUB 2220
1350 GOTO 1430
1360 REM
1370 CALL HCHAR(V,W,136)
1380 GOSUB 1810
1390 IF E=0 THEN 1540
1400 IF M()H THEN 1420
1410 GOSUB 1810
1420 IF E=0 THEN 1540
1430 CALL HCHAR(V,W,32)
1440 GOSUB 2700
1450 IF E=0 THEN 1540
1460 IF M()H THEN 1500
1470 GOSUB 2300
1480 CALL HCHAR(U,Z,32)
1490 GOSUB 2700
1500 IF E=0 THEN 1540
1510 GOSUB 2300
1520 CALL HCHAR(U,Z,32)
1530 GOTO 1370
1540 GOSUB 2300
1550 IF B(6))B(13)THEN 1620
1560 IF B(13))B(6)THEN 1730
1570 FOR X=880 TO 110 STEP -
55
1580 CALL SOUND(100,X,0)
1590 CALL SOUND(100,110,30)
1600 NEXT X
1610 GOTO 1190
1620 CALL HCHAR(21,25,152)
1630 CALL HCHAR(21,28,152)
1640 CALL HCHAR(20,25,152,4)
1650 CALL HCHAR(22,25,152,4)
1660 CALL SOUND(300,220,0)
1670 FOR X=220 TO 1020 STEP
50
1680 CALL SOUND(20,X,0)
```

))))))

```
1690 NEXT X
1700 FOR X=1 TO 1000
1710 NEXT X
1720 GOTO 1190
1730 CALL HCHAR(9,25,152)
1740 CALL HCHAR(9,28,152)
1750 CALL HCHAR(8,25,152,4)
1760 CALL HCHAR(10,25,152,4)
1770 GOSUB 590
1780 FOR X=1 TO 1000
1790 NEXT X
1800 GOTO 1190
1810 CALL KEY(0,K,S)
1820 CALL HCHAR(V,W,136)
1830 IF S=0 THEN 1810
1840 IF K<>68 THEN 1900
1850 CALL HCHAR(V,W,32)
1860 W=W+3
1870 IF W<25 THEN 1810
1880 W=24
1890 GOTO 1810
1900 IF K<>83 THEN 1960
1910 CALL HCHAR(V,W,32)
1920 W=W-3
1930 IF W>8 THEN 1810
1940 W=9
1950 GOTO 1810
1960 M=(W-6)/3
1970 CALL SOUND(100,440,1,55
0,1)
1980 CALL SOUND(100,550,1,66
0,1)
1990 M=M-1
2000 IF B(M)=0 THEN 1810
2010 H=6
2020 GOSUB 2040
2030 GOTO 2300
2040 T=M
2050 GOSUB 2550
2060 E=0
2070 IF T<=6 THEN 2090
2080 T=T-7
2090 C=C+1
2100 IF C>=9 THEN 2120
2110 F(N)=F(N)*6+T
2120 FOR I=0 TO 5
2130 IF B(I)<>0 THEN 2160
2140 NEXT I
2150 RETURN
2160 FOR I=7 TO 12
2170 IF B(I)=0 THEN 2200
2180 E=1
2190 RETURN
2200 NEXT I
2210 RETURN
2220 GOSUB 370
2230 PRINT "          UMBALA K
UBUI": : : :
2240 PRINT "HOME";TAB(25);"H
OME":" PIT";TAB(25);"PIT"
2250 PRINT : : : : : :
2260 PRINT :"          ";A$: :
2270 FOR X=12 TO 18
2280 CALL HCHAR(X,8,152,18)
2290 NEXT X
2300 FOR I=12 TO 7 STEP -1
2310 CALL HCHAR(13,9+3*(12-I
),B(I)+96)
2320 NEXT I
2330 FOR I=0 TO 5
2340 CALL HCHAR(17,9+3*I,B(I
)+96)
2350 NEXT I
2360 D1=INT(B(13)/10)
2370 D2=B(13)-10*D1
2380 CALL HCHAR(9,26,D1+48)
2390 CALL HCHAR(9,27,D2+48)
2400 D1=INT(B(6)/10)
2410 D2=B(6)-10*D1
2420 CALL HCHAR(21,26,D1+48)
2430 CALL HCHAR(21,27,D2+48)
2440 IF B(13)>16 THEN 2470
2450 CALL HCHAR(15,5,B(13)+9
6)
2460 GOTO 2490
2470 CALL HCHAR(15,5,112)
2480 CALL HCHAR(15,6,B(13)+8
0)
2490 IF B(6)>16 THEN 2520
2500 CALL HCHAR(15,27,B(6)+9
6)
2510 RETURN
2520 CALL HCHAR(15,27,112)
2530 CALL HCHAR(15,28,B(6)+8
0)
2540 RETURN
2550 P=B(M)
2560 B(M)=0
2570 FOR P=P TO 1 STEP -1
2580 M=M+1
2590 IF M<=13 THEN 2610
2600 M=M-14
2610 B(M)=B(M)+1
2620 NEXT P
2630 IF (B(M)=1)+(M<>6)+(M<>
13)<>-3 THEN 2650
2640 IF B(12-M)<>0 THEN 2660
2650 RETURN
2660 B(H)=B(H)+B(12-M)+1
2670 B(M)=0
2680 B(12-M)=0
2690 RETURN
2700 D=-99
2710 H=13
2720 FOR I=0 TO 13
2730 G(I)=B(I)
```

>>>

```
2740 NEXT I
2750 FOR J=7 TO 12
2760 IF B(J)=0 THEN 3080
2770 Q=0
2780 M=J
2790 GOSUB 2550
2800 FOR I=0 TO 5
2810 IF B(I)=0 THEN 2920
2820 L=B(I)+I
2830 R=0
2840 IF L(=13 THEN 2880
2850 L=L-14
2860 R=1
2870 GOTO 2840
2880 IF (B(L)=0)+(L()6)+(L()
13)()-3 THEN 2900
2890 R=B(12-L)+R
2900 IF R(=Q THEN 2920
2910 Q=R
2920 NEXT I
2930 Q=B(13)-B(6)-Q
2940 IF C)8 THEN 3020
2950 T=J
2960 IF T(=6 THEN 2980
2970 T=T-7
2980 FOR I=0 TO N-1
2990 IF F(N)*6+T()INT(F(I)/6
^(7-C)+.1)THEN 3010
3000 Q=Q-2
3010 NEXT I
3020 FOR I=0 TO 13
3030 B(I)=G(I)
3040 NEXT I
3050 IF Q(D THEN 3080
3060 A=J
3070 D=Q
3080 NEXT J
3090 M=A
3100 Z=3*(15-M)
3110 CALL HCHAR(U,Z,144)
3120 CALL SOUND(100,440,1,55
0,1)
3130 CALL SOUND(100,440,1,33
0,1)
3140 GOTO 2040
3150 FOR I=0 TO N-1
3160 PRINT B(I)
3170 NEXT I
3180 END
```

ACKNOWLEDGEMENT:

This listing was produced from an initial 28 column format output, the product of a program kindly supplied by JIM PETERSON of TIGERCUB SOFTWARE.

This should provide a WYSIWYG effect (What You See Is What You Get) when typing the listing in and checking the screen.

)))))))

MANIPULATING GRAPHICS DEFINITION STRINGS

Originally written for HOME COMPUTING WEEKLY

If you have ever sat down with a sheaf of graph paper and a pencil, to work out the rotating shapes of your favourite Alien Mother-in-law, or to laboriously draw out a sequence depicting a shape moving within the confines of the 8 by 8 dot matrix of a character, then you might eventually have come to the conclusion that life could be simpler if you could get the computer to shoulder some of the work. For the most part, manipulating the hexadecimal definition strings used to describe a shape can be straightforward and quite fast, especially if you use efficient algorithms (the 'rules of thumb' for performing a particular task). It is even possible to produce a range of varying-resolution images through definition string manipulation, although the procedure is too complicated to go into here.

Simple manipulation can quickly give you the 'transformed' definition string for an upside-down version of a given shape, or a left-right mirror image, or an inverse video of either or both. The upside-down transformation is simplest so we'll deal with it first. All it involves is writing the definition string backwards, pair by pair (of hexadecimal digits). Thus "0123456789ABCDEF" will become "EFCDAB89674523301". Working on a maximum of 16 digits, this routine performs the elementary task for you:

```
 10  REM  LISTING No. 1
 20  REM  UPSIDE-DOWN TRANSFORMATION
100 CALL CLEAR
110 CALL SCREEN(8)
120 Z$="0000000000000000"
130 E$=""
140 INPUT "ENTER UP TO 16 HEX DIGITS:":D$
150 D$=SEG$(D$&Z$,1,16)
160 CALL CHAR(128,D$)
170 PRINT ::CHR$(128);" ";D$::
180 FOR L=1 TO 15 STEP 2
190 E$=SEG$(D$,L,2)&E$
200 NEXT L
210 CALL CHAR(129,E$)
220 PRINT CHR$(129);" ";E$::
230 GOTO 130
```

Line 150 ensures that the string is 16 digits long, including trailing zeroes automatically. You could insert additional lines to scan the string and isolate any non-hexadecimal characters, or to give hard copy on a printer.

A left-right mirror image is more difficult, but this routine will do it for you:

)))))))

```
 10  REM  LISTING No. 2
 20, REM  LEFT-RIGHT TRANSFORMATION
 100 CALL CLEAR
 110 CALL SCREEN(8)
 120 Z$="0000000000000000"
 130 H$="0123456789ABCDEF"
 140 R$="084C2A6E195D3B7F"
 150 E$=""
 160 INPUT "ENTER UP TO 16 HEX DIGITS:":D$
 170 D$=SEG$(D$&Z$,1,16)
 180 CALL CHAR(128,D$)
 190 PRINT ::CHR$(128);" ";D$::
 200 FOR L=1 TO 15 STEP 2
 210 E$=E$&SEG$(R$,POS(H$,SEG$(D$,L+1,1),1),1)&
     SEG$(R$,POS(H$,SEG$(D$,L,1),1),1)
 220 NEXT L
 230 CALL CHAR(129,E$)
 240 PRINT CHR$(129);" ";E$::
 250 GOTO 150
```

H$ and R$ are reference strings, and each hexadecimal digit in H$ has its left-right mirror image counterpart in the corresponding position in R$. Thus "8" is the mirror of "1", "4" of "2", etc. The full operation is too detailed to go into here.

The inverse video routine, which switches all ON dots off, and OFF dots on, is similar:

```
 10  REM  LISTING No. 3
 20  REM  INVERSE VIDEO TRANSFORMATION
 100 CALL CLEAR
 110 CALL SCREEN(8)
 120 Z$="0000000000000000"
 130 H$="0123456789ABCDEF"
 140 E$=""
 150 INPUT "ENTER UP TO 16 HEX DIGITS:":D$
 160 D$=SEG$(D$&Z$,1,16)
 170 CALL CHAR(128,D$)
 180 PRINT ::CHR$(128);" ";D$::
 190 FOR L=1 TO 16
 200 E$=E$&SEG$(H$,17-POS(H$,SEG$(D$,L,1),1),1)
 210 NEXT L
 220 CALL CHAR(129,E$)
 230 PRINT CHR$(129);" ";E$::
 240 GOTO 140
```

It should be straightforward to combine all three routines into one larger routine, with a short menu, permitting the operation of any or all routines in combination. It would also be possible to expand the routines so that they become capable of handling far more than just 16 digits, although that kind of programming and graphics manipulation will have more appeal to the seasoned programmer.

Coping with image rotation is quite difficult, and will usually involve a matrix of binary ones and zeroes corresponding to the shape, set in a two-dimensional array, with the necessary processing to turn the shape through 360 degrees in 45 degree steps, or whatever, followed by translation into hexadecimal. It can be slow at times, but nowhere near as slow (and boring) as doing it all by hand!

All of which gives me the chance to go into it all over again in detail next issue...

)))))))

JOHN ROE wrote to me with some observations on the mention of the Calculus of Finite Differences which appeared in V2.11. Basically, if the relationship between two sets of numbers is reasonably simple, (like Y = (2*X+5)/7), then this Calculus will work out that relationship with relatively little head scratching. At some stage there will be either a QUINSOFT booklet/video, or an ITUG BOFFINS SUPPLEMENT covering some of the Maths which can be of use to programmers looking to beef up their knowledge.

In the meantime, John examined what I had done and realised that there was a simpler, more general technique which works better within certain limitations (mostly imposed by the maximum length of a program line).

This is useful feedback, which is always welcome, as it is all too easy to go charging off down some obscure path pursuing an esoteric method of doing something which can be achieved much more simply.

John's version of the initial TAB() expression goes like this:

```
PRINT TAB(-4*(S=0)-15*(S=1)-28*(S=2))
```

This is one of those dreaded RELATIONAL EXPRESSIONS which is worth getting to grips with, as it can be useful in many other circumstances.

Remember that the expression within the brackets (S=...) will be evaluated by the computer to see if it is TRUE or FALSE. A value of -1 is produced for a TRUE result, while 0 is equivalent to FALSE.

For a loop of S = 0 to 2, the first time through the loop in which the above PRINT statement is evaluated, when S=0, the first expression will have a result of -1 (because S=0 is TRUE), and the other two will have results of 0 (because S=1 is FALSE, and S=2 is FALSE). The intermediate result is:

```
PRINT TAB(-4*(-1)-15*(0)-28*(0))
```

When any number is multiplied by zero, the result is zero. So 15*(0) and 28*(0) are both zero. This leaves -4*(-1), and the rules of multiplication of signed numbers state that a negative number multiplied by another negative number yields a positive result. So -4*(-1) is +4, or just 4.

Thus the statement evaluates to the equivalent of PRINT TAB(4) when S=0.

When S is 1 or 2, then (S=0) will be FALSE every time, so -4*(S=0) will always be zero. If S=1, then (S=1) will be TRUE (-1) while (S=2) will be FALSE, and thus zero, reducing the statement to PRINT TAB(-15*(-1)),

))))))

or PRINT TAB(15).

When S=2, only the (S=2) expression will be TRUE, while the others are FALSE (and thus zero), which reduces the PRINT statement to the equivalent of PRINT TAB(28).

Obviously there are limits to this technique, but at least you don't have to learn (or utilise) Calculus!

You may have spotted the pattern already. The numbers in the brackets are the values which the loop will take. Obviously if S looped from 1 to 50 you'd not be able to make use of this technique because you couldn't fit it all in one statement (although the intrepid among you may see that you could split the line up, assigning intermediate results to a variable, and finally using TAB() with that variable).

However, for most simple purposes this is a practical solution.

The required TAB() values are given by the negative numbers used to multiply whatever is within the brackets. Thus the TAB() positions wanted were 4, 15, and 28, so each expression is multiplied by either -4, -15, or -28.

John went on to give a slightly more general expression which could be used as part of an Extended BASIC User-written subprogram. Substituting the variables T1, T2, and T3 for the 4, 15, and 28, gives:

```
PRINT TAB(-T1*(S=0)-T2*(S=1)-T3*(S=2))
```

The CALL to the routine might therefore use:

```
CALL SHOWLIST(T1,T2,T3,L$(,))
```

and the routine might be:

```
1000 SUB SHOWLIST(T1,T2,T3,X$(,))
1010 FOR P=0 TO 19::FOR S=0 TO 2
1020 PRINT TAB(-T1*(S=0)-T2*(S=1)-T3*(S=2));X$(S,P);
1030 NEXT S::NEXT P::SUBEND
```

Or something along those lines.

If you have any further comments, or programming tips which you think might be worth passing on, then send them in for consideration.

))))))

THE TI BASIC SYMBOL TABLE

BOB DILWORTH of the NEW HORIZONS USER GROUP

Variables certainly are a blessing to any programmer. Without them even the simplest loop problem would be well-nigh impossible. If you aren't convinced, try writing the equivalent of a FOR/NEXT loop without using a variable. While it may be a good exercise in logic, a steady diet would quickly fatten up even the leanest and meanest code.

Even though we all take variables for granted, I'm sure we've wondered just where in the computer they're stored, and how the 99/4A knows where they are and how to change them. Let's write a simple little BASIC program that does nothing more than assign some values to two variables. We'll make one a string variable (i.e. a variable that will contain a string of letters), and another a numeric variable (one that will contain numbers).

```
10 OLD$ = "HELLO  THERE"
20 NMBR = 5
```

Type in RUN to register the variables and their values. Now, if we type in PRINT OLD$, the words HELLO THERE will be printed on the next line. Similarly, if we type in PRINT NMBR the number 5 will be displayed.

So what, you may be asking about now. This is low level stuff. I agree, you're absolutely right. It does, however, get a little more interesting when you begin to explore VDP RAM to uncover how the computer stores and acceses its variables.

THE SYMBOL TABLE

The symbol table is the area in VDP RAM where the 99/4A stores the variables assigned during the execution of the BASIC program. In console BASIC, this table begins somewhere after address 0800 (hex). It also dynamically expands and contracts in size depending on what happens to individual variables during the execution of the BASIC program. The address 833E (hex) in the CPU RAM scratch pad area always points to (contains the address of) the beginning of this table. How, then, are the variables stored in this area of VDP RAM? As our example, let's use the short program we previously discussed. The address contained in CPU RAM address 833E (hex) points to the beginning of the entry of the variable most recently interpreted by BASIC. In our example, this would be the variable NMBR. The entry is arranged in the following manner:

```
0004 3735 376A 4005 0000 0000 0000
```

))))))

This seven word (14 byte) area is interpreted as follows:

Word 1 : 0004

The first "0" indicates this variable is numeric. A string variable would begin with an "8". The second "0" indicates that the variable is not subscripted. In other words, it has not been defined as NMBR(1). The second byte "04" indicates that the variable name (NMBR) is four ASCII characters long.

Word 2: 3735

This is an address in VDP RAM where the beginning of the next table entry begins. In our example, it points to the table entry for the variable OLD$.

Word 3: 376A

This is an address in VDP RAM where the actual string (NMBR) is located.

Words 4-7

This is the number 5 in RADIX 100 notation. RADIX 100 notation is used by the computer to represent floating point numbers. A semi-useful discussion of it can be found on page 279 of the Editor Assembler manual.

The second entry immediately follows the first, and is arranged in the following manner:

8004 0000 3775 370A

As you can see, this entry is much shorter than the previous one. It breaks down as follows:

Word 1: 8004

The first, "8", indicates that this is a string variable. The second, "0", indicates that the variable is not subscripted. In other words, it has not been defined as OLD$(1). The second byte, "04", indicates that the variable name (OLD$) is four ASCII characters long.

Word 2: 0000

))))))

Link to the next table entry, if there is one. The zeros here indicate that this is the last variable in the table.

Word 3: 3775

Address in VDP RAM where the actual string of the variable name "OLD$" is located.

Word 4: 370A

Hex address in VDP RAM where the current contents of the variable "OLD$" are located. In this example the address points to the beginning of the string "HELLO THERE".

All together, then the symbol table entry would appear exactly as:

0004 3735 376A 4005 0000 0000 0000 8004 0000 3775 370A

SUBSCRIPTED VARIABLES

Subscripted and multi-dimensioned variables such as in DIM A$(2,1,4) are similarly arranged, but unfortunately, it would take too much valuable space to enter into a complete discussion of their symbol table entry layout. Why don't we reserve that discussion for another time, and concentrate right now on simple numeric and string variables.

CHANGING THINGS

With this knowledge, we can access these areas in VDP RAM via an Assembly Language routine and really change things around. Such a routine is presented at the end of this article. The remaining discussion will deal directly with an explanation of how the routine works, and how to call it from TI Console Basic.

THE REDO ROUTINE

This program will demonstrate how to access the symbol table discussed above. It will change both the variable name and contents of one string and one numeric variable registered via a TI Console BASIC program. The Assembler routine is written for the Editor/Assembler, and runs via a CALL LINK statement from BASIC.

Using the Editor/Assembler, type in the program as it is reprinted below. Assemble the program, and call the object code DSK1.REDO/O. Leaving the Editor/Assembler cartridge plugged in, enter TI BASIC and type in the following BASIC program:

```
10 OLD$ = "HELLO THERE"
20 NMBR = 5
30 CALL LOAD("DSK1.REDO/O")
```

))))))

RUN the program.

If you now type in PRINT OLD$, the string "HELLO THERE" will be displayed. Typing in PRINT NMBR displays the number "5". To run the REDO program, simply type in CALL LINK("REDO"). The REDO program will change the variable name OLD$ to NEW$, and change the contents of the variable from "HELLO THERE" to "GOODBYE NOW". The variable name NMBR will be changed to WXYZ, and the contents from "5" to 10.26070801. The old variable name OLD$ and NMBR are no longer registered in the symbol table. Typing in PRINT NEW$ and PRINT WXYZ will display the new variable contents. Typing in PRINT OLD$ will display a line of spaces: typing in PRINT NMBR will display a "0".

```
        AORG >3000
        DEF  REDO
        REF  VMBW,VMBR
WKSP    BSS  32
RETURN  BSS  2
SYMTAB  BSS  2
LENHLD  DATA >0000
NEXT    BSS  2
NAME    BSS  2
STRNG   BSS  2
NAMLEN  DATA >0000
TWO     EQU  >0002
NWNAME  TEXT 'WXYZ'
NEWNUM  DATA >400A,>1A07,>0801,>0000
STNAME  TEXT 'NEW$'
NEWST   TEXT 'GOODBYE NOW'
**
**
REDO    MOV  R11,@RETURN      " SAVE RETURN ADDR IN R11
        LWPI WKSP             " LOAD OWN WORKSPACE
        MOV  @>833E,R8        " START OF NUMERIC VARIABLE TABLE ENTRY TO RO
        BL   @SUBRTN          " GO TO SUBRTN TO REPLACE VARIABLE NAME
        MOV  R8,R0            " ADDR OF THE NUMBER "5" IN RADIX 100 TO RO
        LI   R1,NEWNUM        " NEW RADIX 100 NUMBER ADDR
        LI   R2,8             " 8 (FOR 8 BYTES) TO R2
        BLWP @VMBW            " REPLACE OLD NMBR CONTENTS WITH NEW
        MOV  @STNAME,@NWNAME  " MOVE STRING VAR NAME TO NWNAME
        MOV  @STNAME+2,@NWNAME+2
        MOV  @NEXT,R8         " START ADDR OF STRING VAR ENTRY TO R8
        BL   @SUBRTN          " GO DO SUBRTN TO REPLACE THE VAR NAME
        MOV  R8,R0            " ADDR OF BEGINNING OF STRNG CONTENTS
        LI   R1,STRNG         " ADDR  OF HOLDER FOR THIS
        LI   R2,2             " TWO BYTES LONG
        BLWP @VMBR            " GO GET VAR STRING CONTENT ADDR FROM TABLE
        MOV  @STRNG,R0        " ADDR OF OLD VARIABLE STRING TO RO
        LI   R1,NEWST         " ADDR OF NEW VARIABLE STRING TO R1
        LI   R2,11            " ELEVEN BYTES LONG
        BLWP @VMBW            " REPLCE OLD WITH NEW STRING IN VDP RAM
        LWPI >83E0            " LOAD GPL WORKSPACE
        MOV  @RETURN,R11      " RESTORE RETURN ADDR
        B    *R11             " BYE!!
*
* SUBROUTINE SECTION *
*
SUBRTN CLR  @LENHLD           " CLEAR OUT HOLDING AREAS

))))))
```

```
       CLR  @NAMLEN
       MOV  R8,RO            " R8 HAS ADDR OF LEN OF VAR NAME
       LI   R1,LENHLD        " ADDR OF HOLDER FOR NAME LENGTH TO R1
       LI   R2,2             " ADDR IS TWO BYTES LONG
       BLWP @VMBR            " READ IN LENGTH  OF VAR NAME TO HOLDER AREA
       SWPB @LENHLD          " SWAP IT TO MOST SIGNIFICANT BYTE
       MOVB @LENHLD,@NAMLEN " MOVE LENGTH TO NAMELENGTH HOLDER AREA
       SWPB @LENHLD          " PUT IN LEAST SIGNIFICANT BYTE
       AI   R8,2             " ADD 2 TO THE ADDR IN R8
       MOV  R8,RO            " MOVE IT TO RO
       LI   R1,NEXT          " ADDR OF HOLDER FOR NEXT LINK IN S TABLE
       LI   R2,2             " IT'S 2 BYTES LONG
       BLWP @VMBR            " GO GET IT FROM VDP RAM
       AI   R8,2             " ADD 2 TO ADDR IN R8
       MOV  R8,RO            " STASH IT IN RO
       LI   R1,NAME          " HOLDER AREA FOR ADDR OF VARIABLE NAME
       LI   R2,2             " READ IN TWO BYTES
       BLWP @VMBR            " GO GET IT
       MOV  @NAME,RO         " MOVE ADDR OF VARIABLE NAME TO RO
       LI   R1,NWNAME        " ADDR OF NEW VARIABLE NAME TO R1
       MOV  @NAMLEN,R2       " LENGTH OF NAME TO R2
       BLWP @VMBW            " REPLACE IT
       AI   R8,2             " ADD 2 TO ADDR IN R8 TO POINT TO NEXT TABLE
       B    *R11             " RETURN TO MAIN ROUTINE
       END
```

))))))

"On the topic of the 32K RAM Expansion, when I first built mine it did not work. This was traced to a bad contact on the HIMEM)EF chip. From this I concluded that for Extended BASIC to operate it was necessary to have all four memory chips in the system.

However, this now does not appear to be the case. It is essential to have LOMEM fitted as this is used for machine language subprograms. The Extended BASIC Operating System stores programs and data starting from the top of memory and working down. This is why my unit did not work with a faulty HIMEM)EF chip. However, I have been informed that if the LOMEM)AB is left out, the system still works, except that when you use the SIZE command you get a maximum 16K of program space free instead of 24K. It may therefore follow that if you also omit HIMEM)CD the system would still work but would only allow 8K of program space.

I have not tried this myself so I would appreciate any feedback from anybody who tries it. If it proves to be correct (and it is not certain that the system would function 100% with less memory) it would allow people who don't require the full expansion to simply plug in what they require, and then if later they require more memory they can always add the extra ICs.

If anyone pursues this idea further I hope that they will write about it so that we can all benefit.

I have also managed to repair an apparently dud console, and the details may be of help to others with similar problems.

The title screen would appear but the screen colour was wrong and several of the characters were incorrect. If I proceeded into TI BASIC I could enter a line but a lot of the keys produced wrong characters and even when a correct line was entered the machine gave an unintelligible error message. The wrong characters and screen colour seemed randomly generated, being different each time the console was turned on.

I was certain that the processor was working otherwise the machine would not work at all, so my first thought was that it was either the VDP IC, one of the ROMs or GROMs or one of the RAM ICs. The first check was the VDP since this was socketed. Plugging in a spare did not cure the fault.

The next line of attack was to check the RAM. The TI-99/4A uses eight TMS4116 16K by 1 RAM ICs which are unfortunately all soldered into the double-sided plated board. The first task was to remove them all and refit them with sockets. To prevent damage to the board I used a special soldering iron bit that desolders all 16 pins at once and then used solder wick to remove the surplus solder from the holes.

It is then a simple job to fit sockets and replace the ICs. I tried the console again to confirm that the fault was still the same and that I had not created any further faults.

Having borrowed a spare IC from a friend I tried substituting it for each of the eight RAM ICs in turn until I found a faulty one. Once I had replaced this all was well.

I now have a fully working spare console with the bonus that the memory is all in sockets so future fault finding should be easier. I hope that this information may be of use to anyone who has experienced a similar fault."

))))))

COMMENTS

By COLIN HINSON

PIO, CENTRONICS AND ALL THAT

There are a multitude of different ways of communicating between computers and peripheral devices via an eight bit parallel bus. Some of these use a 'handshake' between the computer and the peripheral for each byte sent, some use only a 'shut up, I can't take any more' (Busy) signal, and some use a combination of both. The hardware on the TI card is capable of dealing with all of these, however the software in the DSR ROM is not, - it is only capable of dealing with the first type (handshake between each byte), which causes peripherals which use a standard 'Busy' signal to 'hang-up'. (Can be cleared by 'CLEAR' key).

The sequence of events when a byte via the DSR ROM 'PIO' software is as follows:

1. Enable the send data bus latch
2. Test 'Busy' (pin 10) input (see note below on pin no.)
3. Wait until 'Busy' is low (not Busy)
4. Send the byte to the data bus latch
5. Set the clock (strobe) output low (pin 1)
6. Wait for 'Busy' to go to a 1 (Busy)
7. Set the clock (strobe) output to a 1
8. Return to caller

This routine was apparently designed to cope with the rather weird handshake used by the 99/4A (Epson) printer and also with those printers which used 'Acknowledge' for a byte by byte handshake. The 'Acknowledge' signal has generally fallen into disuse, as it was only necessary for slow receive systems but note that the general convention is that Acknowledge comes out of the printer on pin 10 and Busy on pin 11 which will usually involve some cross wiring on the cable to get Busy onto pin 10 at the computer end. Note also that most devices using Acknowledge correctly use 'Acknowledge bar' (i.e. signal is normally high and goes low to acknowledge receipt), which the 99/4A DSR will not cope with.

Most printers nowadays use only the Busy signal. This signal is normally low (not Busy) unless the print buffer is full, the device is off line, out of paper etc., etc., and is normally called 'true Centronics' (if the output driver at the printer end is a low power schottky device it may also be 'not Busy' when the printer is switched off). As this system produces no change in the Busy signal with each byte, the computer will hang-up waiting for the receipt of a handshake unless the DSR ROM software is changed, the hardware is changed, or you write your own routine to 'talk' directly to the PIO (a sample routine is given later). DSR ROMs with the appropriate modification made to the software are available (talk to Peter Brooks). These ROMs are in EPROM form and these require that the VPP pin (pin 21) is wired to VCC (pin

))))))

24) so if you are fitting one of these EPROMs ensure that pin 21 is bent under the device and connected to pin 24, - do not attempt to insert pin 21 on the EPROM into the board as the board has pin 21 wired to ground. Other than this, the EPROM is pin for pin compatible with the 'masked' ROM normally supplied with the card.

If you want (or need) to drive the PIO direct then the following Assembly Language routine will send bytes to the PIO with a 'true Centronics' Busy signal from the printer:

```
        LI    R12,>1300         POINT TO RS232 CARD
        SBO   0                 TURN ON ITS DSR ROM (AND LATCH GATING)
        SBZ   1                 ENABLE THE TRANSMIT BUS LATCH
*
BUSYLP  TB    2                 TEST BUSY (PIN 10)
        JEQ   BUSYLP            LOOP TILL NOT BUSY
        MOVB  @SEND,@>5000      SEND BYTE TO BUS LATCH
        SBZ   2                 SET STROBE TO ZERO
        SBO   2                 SET STROBE TO ONE
*
        SBO   1                 TRISTATE THE BUS LATCH
        SBZ   0                 TURN OFF THE DSR ROM
```

The byte at location 'SEND' can of course be addressed in any of the normal addressing modes of the TMS9900. If multiple bytes are to be sent across the PIO system without having to access any other DSR ROMs (such as the disk controller) then only the 5 lines of code between the two '*'s need be executed during this time, but the RS232 ROM must be turned off before another is enabled or the ensuing fight for the bus will give you a 'Device not found' error.

****

CARTRIDGE PORTS

Are you having trouble with your Cartridge Port? Does your computer crash for no apparent reason, but work OK when you switch off and on again? - Well the chances are that the connector assembly needs servicing! When you first got your computer the surround of the connector itself had a fluid in the felt (or foam plastic depending on which version). This fluid is a contact lubricant (NOT a contact cleaner), and is designed to allow the port contacts to give good contact onto the cartridge edge connector (the felt (or foam) does the cleaning by pushing the dirt to the back of the cartridge edge connector). After a lot of use (or a lot of time), the fluid will either be used up or dry out (usually about 2 years with average use).

The fluid used is an equivalent of 'Electrolube Plus 2X' fluid, and comes in a dispenser which looks like a fat fountain pen and has a long nozzle. This fluid should be applied to the felt until it won't absorb any more. To gain access to the connector, dismantle the computer, remove the main board (vertically to avoid bending the cartridge connector). Unplug the connector from the main board, and then unclip the piece of plastic (which holds the foam) from the connector. Apply the fluid to the felt and a small amount to the edge connector which

>>>

plugs into the main board and then re-assemble. Make SURE that the piece of support metal below the connector is central on it and that its lip is in front of the connector (looking from the cartridge end) as failure to do this will make cartridge insertion difficult (or if the lip is below, impossible).

NOTE: DO NOT, REPEAT NOT, USE AN AEROSOL TYPE CONTACT CLEANER (EVEN IF IT IS 'ELECTROLUBE')

****

32K RAM EXPANSION

I know it's a bit late now, but I have only just had the copies of TI-LINES which refer to the external 32k RAM expansion.

There really is no need to do any wiring to the '6 way Din' at all surely? If the TI PEB can do without it then it follows that this project can, as all the signals required will be available on the I/O port (including the +5v).

What is required is to produce:

4 chip selects (1 for each RAM)
1 inverted DBIN signal.

The inverted DBIN signal can be produced by using any TTL device which inverts (such as an LS04, LS00, LS02 - whichever you happen to have), and the 4 chip selects can be produced by using an LS138 decoder (i.e. duplicating the circuit on the main board from which the wires were taken). If you don't have a circuit of the main board (they are available from T.I. still!), then wire up the LS138 as follows:

138 Pin	Name	to I/O pin	name	Remarks	
1	A	20	A2		
2	B	30	A1		
3	C	31	A0		
4	G2A	32	MEMEN BAR		
5	G2B	32	MEMEN BAR		
6	G1	1	+5V		
7	Y7			CS1	()E000-)FFFF)
8	GND	21,23,25,27	GROUND		
9	Y6			CS2	()C000-)DFFF)
10	Y5			CS3	()A000-)BFFF)
11	Y4	NOT USED			()8000-)9FFF)
12	Y3	NOT USED			()6000-)7FFF)
13	Y2	NOT USED			()4000-)5FFF)
14	Y1			CS4	()2000-)3FFF)
15	Y0	NOT USED			()0000-)1FFF)
16	VCC	1	+5V		

It does not matter really which chip select you take to which RAM. (If it comes to that, it doesn't matter which addresses (A3-A15) you take to which RAM address pins, or data bits you take to which RAM data

))))))

pins on the RAMS as you will only get out of the RAM what you put in - its internal address is irrelevant)

****

IN PARTIAL DEFENCE OF 'WHEN THE DISK MANAGER IS SLOW' (V2.8)

First of all, let us consider the way in which the disk controller works. When the controller chip is reading data off the disk, the data comes to it in serial form, i.e. one bit after another interspersed with 'clock' bits. The time taken to read an 8 bit byte is approximately 25usecs which is of course far longer than it takes the 9900 to read the byte from the disk controller. In order to ensure that the byte is ready for the 9900, the 'Ready' signal from the disk controller chip is used. This Ready (Wait) is only enabled when the 9900 needs to read or write data to the disk (by a signal called 'Waiten' (Wait Enable)), otherwise the 9900 would be waiting for ever when doing such things as reading the DSR ROM or writing to the track registers etc.

So, you have inserted a disk which you wish to initialize, and it is un-initialized. The 9900 commands the controller to step to track 0, and then enables the 'Waiten' and attempts to read sector zero. As there is no data on track zero, a byte will never be read by the disk controller, and so 'Ready' will never become ready. This would appear to produce 'lock up', however there is a monostable (timer) on the disk controller board which is triggered before each operation by the 9900. When this times out, it automatically disables 'Waiten' and so the 9900 lock up is removed - the timing of this monostable needs to be such that it will more than cover the time taken to step the head from the inner track to the outer track on an eighty track drive. This time is the time you have to wait for the Disk Manager to say 'disk not initialised'.

There is of course a way round having to wait while the controller finds out that the disk is not initialised, and that is to initialize it without reading sector zero (this is done with the 'Box Format' in DM1000) - however this is only a good idea when you have a boxful of disks to format. I might add that the fact that the disk name appears on the screen has on more than one occasion saved me from initialising the wrong disk!

Finally, may I point out that it is always possible to improve upon somebody else's software, but the improvements may only be improvements to YOU, you may not be taking account of all contingencies, and you may be removing some of the 'Idiot proofing' built into the original.

****

CORRECTING MISLEADING INFORMATION

V2.7 (page 7) contains an article on fitting more than one module GROM in a single module. I should like to point out that pin 14 of a GROM is not the reset line, but is the -5v line (the reset line is on the OTHER end of the 100 ohm resistor (or in early modules a shorting link)). What happens is that when the module is plugged into the

))))))

cartridge slot, the -5v is applied (via the 100 ohms or 0 ohms) to the reset circuit. This discharges a capacitor which produces a pulse into the reset of the 9900 (via the LS362) and so causes the console to reset. T.I. GROMs are 'PMOS' (as opposed to NMOS) devices and so require a negative supply. When this supply is removed as described in the article, all the transistors in the GROM will turn off and so the GROM will not be activated when selected. Some GROMs (e.g. those made by Parker Bros (of Popeye fame) are NMOS devices and do not require a negative supply, consequently selection of the GROM by switching the negative supply will not work with these (as you will notice if you plug a Parker Bros cartridge into a Wiget along with some other cartridge).

Same issue, page 9 - Pseudocode is generally taken to be any code (usually a byte), which has to be INTERPRETED by the microprocessor in order to run. Hence all Interpretive languages such as some Pascals, the better Basics (including the 99/4A), GPL, etc use Pseudocode in order to reduce the amount of memory required to store the program. One of the reasons for TI Basic being so slow is that some of the Basic Interpreter is written in GPL which of course has to be Interpreted itself!

[I have spoken to Colin about the process of Pseudocoding, which, in the form which I presented in the newsletter, Colin had not come across in some 15 years in the industry. The principle is exactly the same: you are producing something which is not to be executed, but interpreted, except that instead of the processor doing the interpreting, YOU do - by translating your personal pseudocode into one of the available high or low level computer languages. Part of my source of information was PROGRAMMING IN BASIC, a book written by JOHN COOKSON and IAIN RICHMOND, the former from the London Hospital Medical College, the latter from the Edinburgh Regional Computing Centre. It does appear that the term pseudocode has been expanded to include environments outside the domain of the computer processor. PB]

))))))

These are the facts. I use different spellings of my address in different ads, so I know that in the last two months my catalogue requests and direct orders have been 43 from the Computer Shopper ads, 37 from the Micropendium ads, and 20 as a result of newsletters, BBS publicity, word of mouth and everything else combined!

My newsletter is being reprinted every month in about 40 newsletters. Other User groups tell me they are reproducing it and passing it out at their meetings. It is being placed on some BBS's. The officers and editors of many User groups have done everything they could think of to help my business - a couple even sent me cash contributions, a few others reprinted my entire catalogue in their newsletter. I was voted a lifetime honorary member of one group, etc., etc. I do greatly appreciate all of this - but it hasn't done much good!

The reason I am writing now is because I have been receiving a few requests that I take out paid ads in U.G. newsletters, and one letter suggesting that I had a responsibility to take out advertising, and to offer substantial discounts as well, to the User groups who were "keeping my business alive"(!)

Now a $5 ad in each of 120 newsletters would cost me more than my net profit in an average month - and I have no reason to think that it would bring me any business whatsoever!

However, I will make an offer. If anyone sends me an order before 1 January 1986 and mentions the name of his User group, next January I will send directly to that User group a rebate of 10% of the order, not including PP&M charges (the PP&M portion of postpaid full disk collections is $1.50). This is not a discount to the individual, it is a payment to the User group for whatever advertising, promotion or support that you choose to give me.

[Naturally, this offer has long since lapsed, but it is worth including the details for the reader's information. PB]

I am not suggesting that you have any obligation to support my business. I know that many of you are only interested in business and financial programs, which I do not offer. I know that most of the User groups have large libraries of Public Domain software - although some of you charge your members quite a bit for those programs. I also know that too many of you get all the software that you want by piracy.

That is why most of my business is coming from those who do not belong to User groups, who do not have access to all the Public Domain and Freeware and piracy. I put some of my Basic programs on cassette in a local retail store, and sold more cassette programs in that one store than I did by mail order worldwide.

It seems that it is time that I divert my efforts to finding those who still want what I have to offer.

))))))

P R O J E C T N O T E S

4-GING ahead is the name of the game, and you may recall that in V2.12 of TI-LINES I publicised the NEW HORIZONS RAM DISK.

Since the resubscriptions began to appear, several have indicated a serious interest in the RAM DISK.

This is by way of further information, to let you know how things are going.

4 YOUR IN 4-MATION:

I am setting a target price of £50 per bare board, to include the cost of currency conversion, import duty (if any), and any carriage involved. One of our members is going to the States shortly, and may be able to bring back a number of boards with him and thus keep costs down.

I will need a fairly swift response if the go-ahead is given, so have your clinky oncers ready. If you haven't decided yet, make up your mind quickly as there will be little time for indecision. If you don't buy along with one bulk order, you may have to wait a couple of months for the next.

What happens once we get the boards is a matter for you to decide. We can arrange to build up your board or leave you to do it as and when you wish/can. Remember that bulk orders of electronic components can sometimes save quite a bit of cash, if that helps you to decide.

If I manage to keep well within the £50 target you will be reimbursed where necessary. Remember that ITUG is not intending to make vast profit from the RAM DISK purchase.

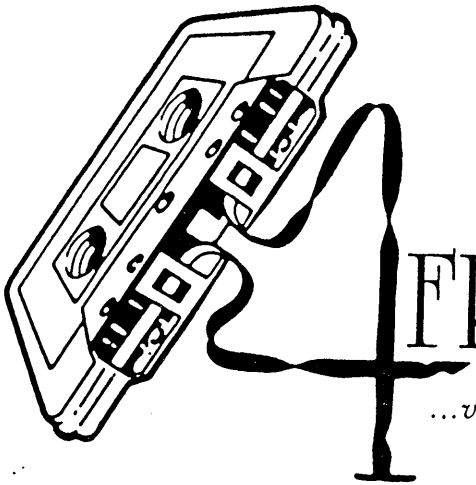
If you have so far indicated that you are "interested" in a RAM DISK, make sure that you amend that to "definitely purchasing" and notify me as quickly as possible, or you may miss the boat - literally.

I have been advised that the RAMDISK is compatible with the MYARC disk controller, and that it should be expandable to a maximum of 1Megabyte.

I will warn every prospective purchaser now that there may be a long delay between sending me the money and receiving the boards. You should be under no illusions that everything will be completed in the twinkling of an eye. Or even two eyes.

As they say, Watch This Space...

PETER BROOKS)))))))



new

4FRONT

...very impressive...
-S.Shaw

Have you subscribed to 4FRONT yet?
If so, this is what you've got:-
ARTICLES/REVIEWS/ADVENTURER'S TIPS/
GREAT GAMES/UTILITIES/FUN & FROLICS
(that was FRolics)/and more besides
for basic, extended basic, assembly
language and PRK users and this lot
all for 3.50 on tape - 4.50 on disk
or a year sub(4) for the price of 3
If not, that's what you're missing.
By the way, why not send an SAE for
our latest pricelist which includes
° SHAMUS ° PROTECTOR ° DEFENDER °
at the most ridiculous prices.....
(plus some new titles on disk/tape)

N E W D A Y C O M P U T I N G
JERRARD CLOSE,HONITON,DEVON EX148EF

New Day
COMPUTING

TEL
0404-
41856

TEXAS SPECIAL!

M O D E M R E V I E W

By GORDON PITT of the WEST MIDLANDS TI USER GROUP

GEC TELECOMMUNICATIONS LTD

DATACHAT 1223A, 1223B MODEM REPORT

In January I phoned GEC and enquired about the possibility of obtaining one of their DATACHAT 1223 modems for the purpose of review for both my own group and for ITUG (formerly OXON TI USERS). I would like to thank Mr R. A. BADMINTON and Mr JULIAN S. LOBLEY in particular for their help and advice, and in fact everyone that I spoke to at GEC were extremely helpful - a rare thing these days.

Towards the end of January the modem appeared late one night, thanks to a neighbour who took it in for me. I eagerly unpacked the box to see what I had in store. I found exactly what I had been promised: a smart slim, self-contained modem requiring no additional power supply - it was even a colour match for my phone!

The handbook came as a bit of a surprise, as these usually result in a fair bit of head scratching, so full marks to GEC for having produced something comprehensible.

I had thought that there was bound to be a problem with the RS232 cable, but to my delight the wiring diagram supplied was straightforward and no trouble at all. The 25 pin D connector was fitted to the TI RS232 card, and the other end, with its 5 way DIN connector, was fitted to the rear of the Datachat. A cable is supplied to connect the modem to the BT socket normally occupied by the phone, and the phone is connected to the modem via another socket at the rear. Power for the Datachat is taken from the phone line itself, so there is no messing around with additional boxes of power supplies and the attendant spaghetti. (Have you noticed how phone sockets are always at least half a mile away from the nearest domestic power point ?).

The operation of the modem was equally straightforward. All I did was to pick up the phone to make sure that I had a dialling tone and then followed the screen instructions given by VIDITEL. I dialled the number I had been given, and once the connection was made I heard the high pitched tone which is known as the carrier signal. On the front of the Datachat are two buttons, two LEDs, and markings indicating what does what. I made sure that the RX/TX (Receive/Transmit) button was in the depressed (RX) position (a bit like the Alphalock key on the 4A console) ready to receive the carrier signal and any subsequent data.

The other button selects whether the modem is 'on-line', and once the 'carrier detect' LED lit up I then pressed the on-line button to complete the connection between BT line, modem, and TI.

After about half an hour wandering around the free Micronet pages I logged off and disconnected by simply releasing the on-line button.

))))))

As you will have read in a previous issue of TI-LINES, TREVOR DAVIES and I took the modem across to Old Baldie's place, where RICHARD SIERAKOWSKI helped wire it up to Peter's system, and we spent a confusing couple of hours looking at many other Bulletin Boards - the confusion, I hasten to add, being caused by the boards and not the modem, which performed faultlessly throughout.

I found the Datachat to be simple and easy to use, and probably a good choice for the newcomer to modem use. It supports 1200/75 and 75/1200 as well as 1200/1200, although this latter must be done under software control. Although I would have liked to have seen a 300/300 rate as well, it is worth noting that in terms of time spent connected via the phone the faster rate (1200) will enable you to receive data in a quarter of the time taken at 300 baud, and thus keep your phone costs down. With the high cost of connection time in the UK (compared with other countries), this is important. The cost on average per local call during off-peak times is about 40p an hour.

Overall I give the unit top marks for operation, instructions, construction, and User-friendliness.

The 1223A model includes all of the above features and adds auto-answer and auto-disconnect facilities.

With the modem you currently receive a free subscription to Microlink, although as that is more business-oriented it is probably not a selling point as far as we are concerned.

I have managed to secure a special price to members of the groups of £65.70 plus post, packing and insurance plus VAT, subject to a minimum order of 25 units. I have so far not received many enquiries, so bear in mind that ITUG intends to put up a 24 hour Bulletin Board later this year and you can prepare to make the most of its facilities by buying your modem now and becoming familiar with the techniques of accessing Bulletin Boards.

[The future lies with information access via the phone (E-mail, and so on), and experience gained now will be of benefit later. It is also possible to access Bulletin Boards in other countries, and to up- and down- load software and other useful items with a modem. PBJ

Elsewhere you will find copies of the GEC information sheet on the Datachat.

If you have any enquiries about the Datachat, contact GORDON PITT on 0922 476373, or at 259 Sneyd Lane, Bloxwich, WALSALL, WS3 2LS.

GORDON PITT)))))))

DATACHAT 1223 KEEPS YOUR MICRO IN TOUCH

Get yourself and your micro into the data communications age with the new GEC Datachat 1223 modem.

Datachat gives you access to a huge amount of information held on databases such as Prestel and Micronet 800.

It lets you communicate through electronic mail systems like Telecom Gold and join in bulletin board conversations right across the country.

It means you can chat and swap programs directly with other micro users.

And it allows you to control your bank account, pay bills and check statements, as well as ordering and paying for goods through electronic shopping systems.

The Datachat 1223 simply connects between your computer and telephone socket. It turns your micro into an information retrieval system, an interactive communications terminal, a finance management centre, an ordering computer for a wide range of goods and services, and a great deal more besides.

Datachat is cheap to use, too. Off-peak local calls can cost as little as 40p an hour*.

And because Datachat 1223 is powered from the telephone line, there's no mains supply or batteries to pay for.

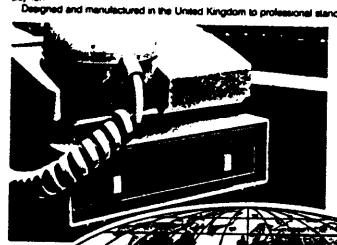
Designed and manufactured in the United Kingdom to professional standards by GEC Telecommunications Limited, Datachat 1223 is ideal for both business and home users.

In technical terms, it's compatible with the international CCITT V.23 standard, providing 75/1200 bit/sec communication. And it's fully approved by the British Approvals Board for Telecommunications.

Your dealer will be able to tell you whether your computer is compatible with Datachat 1223, and if you'll need any communications software.

So go along and talk to your dealer soon, and get into the data communications age.

*Prices correct at time of going to press



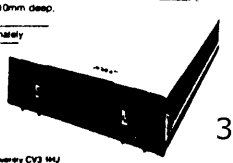
Micronet 800 is the trading style of Telemap Limited and British Telecommunications plc
Prestel and Telecom Gold are registered trademarks of British Telecommunications plc

PUT YOUR MICRO ON THE PHONE

SPECIFICATION

	Modem 1223		
Line Interface			
CONNECTOR — type	A standard BT '600 series' wall socket		
Power	Line powered. Line current is provided by a standard telephone line		
Data Rates			
TRANSMIT MODE —	Transmit 1200 bit/s Receive 75 bit/s		
RECEIVE MODE —	Transmit 75 bit/s Receive 1200 bit/s		
	In accordance with CCITT recommendation V23 1200/1200 bit/s pseudo full duplex is possible under control of the terminal software		
Data Interface	In accordance with CCITT recommendation V24 and RS 232C		
CONNECTOR — type	5-way 240° DIN connector		
— pin connections	CCITT DESIGNATION	RS 232 DESIGNATION	PIN NUMBER
— common return (GND)	102	AB	3
— transmitted data (Tx D)	103	BA	1
— received data (Rx D)	104	BB	5
— request to send (RTS)	105	CA	2
— data channel received line signal detector (DCD)	109	CF	4
Transmitter			
FORMAT	Asynchronous frequency shift keyed modulator (FSK)		
LEVEL	−9 dBm maximum into 600 ohms		
Receiver			
FORMAT	Asynchronous frequency shift keyed demodulator (FSK)		
SENSITIVITY	0 to −43 dBm in any mode		
Frequency Allocation	Binary 0 2100 Hz Binary 1 1300 Hz	at 0 — 1200 bit/s	
	Binary 0 450 Hz Binary 1 390 Hz	at 0 — 75 bit/s	
Data Channel Received Line Signal Detector	'ON' at > −43 dBm 'OFF' at < −48 dBm		
Software Mode Control			
REQUEST TO SEND	'ON' transmit mode 'OFF' receive mode		
Dimensions	140mm wide, 210mm deep, 45mm high		
Weight	0.8 kg approximately		

UK PATENT APPLN NO 8501455



GEC
DATACHAT
MODEM
1223

Modem Division Sales
Transmission Group, GEC Telecommunications Limited, PO Box 53, Coventry CV3 1HJ

STILL AVAILABLE FROM T.I.!

Information supplied by COLIN HINSON and confirmed separately by J. BAPTISTE

Note that some items like the PEB Technical Data manual are still available from ITUG at lower prices (while stocks last).

MANUALS: (While stocks last)

T.I. WRITER	£7
MULTIPLAN	£7
EDITOR/ASS	£9.95
T.I. LOGO II	£7

(NOTE: NO EXTENDED BASIC MANUALS LEFT)
OTHER MODULE MANUALS: £1.15

Prices Include Post and Packing (No VAT)

Write to:

Texas Instruments Ltd,
European Consumer Division
Block C
Manton Centre
Manton Lane
Bedford

If you wish to check that what you want is still available, then ring:

Bedford (0234) 67466, and ask for:

"Consumer Relations, ECD"

Also available (no shortage), are Modulators (£23) and Adaptors (the bit between the Mains plug and the Console) at £12.

Address as above, prices include VAT and P&P.

- - - - - - - -

Circuit Diagrams and Technical Manuals

Circuit diagrams for the whole computer, including the PEB, RS232, Disc Controller and RAM expansion, plus the '99/4a Console and Peripheral Expansion System Technical Data Manual' are available for £8.97 (inc P&P) from:

Leslie Hales
Texas Instruments Ltd
Bedfordia House,
Prebend Street
Bedford.

Also available is a large range of books (Non T.I. authors).
(Note that circuits for the stand-alone peripherals are not available)

>>>

B U L L E T I N B O A R D

WANTED / 4 SALE / WANTED / 4 SALE / WANTED / 4 SALE / WANTED / 4 SALE /

MARK LEE has FOUR BRAND NEW SPEECH SYNTHESISERS for sale, at £20 EACH.

Contact me (BALDIE PETE BROOKS) on OXFORD 50822.

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

DAVE HEWITT has about 8 composite video generators (colour) to enable you to connect your TI to a colour monitor via the standard modulator. Price is £5 each exclusive of post and packing.

There is also a good condition console for sale at £40, and a couple of cartridges: Munchman, and Parsec, both at £3. These prices do not cover post and packing.

Dave is also looking for a DISK CONTROLLER CARD at a reasonable price.

Contact me in the first instance.

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

DAVID BAINES is looking for any documentation for PROGRAMMING AIDS 2. Contact me in the first instance.

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

I still have the Smith Corona TP-I daisey wheel printer for sale, and by the time that you get to read this it should have been fitted with a tractor-feed unit. This will enable it to take either single sheet or fanfold paper, and I'm still only asking £100 for it. I will undertake to deliver it as well, unless the distance is too great (over 100 miles) in which case I will send it DATAPOST. Supplied with cable to connect to RS232/1 on the TI RS232 card, and four extra print wheels.

I am on the lookout for a reasonably-priced TI RS232 card. If you have a spare knocking about, give me a ring or drop me a line quoting the price you want, and I will cough and splutter and explain that I have a vast family to support and then offer you half what you expect me to.

That's Baldie Pete Brooks, in case you hadn't guessed...

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

IT COSTS NOTHING TO ADVERTISE ON THE BULLETIN BOARD

EXCEPT FOR THE PRICE OF A STAMP

))END